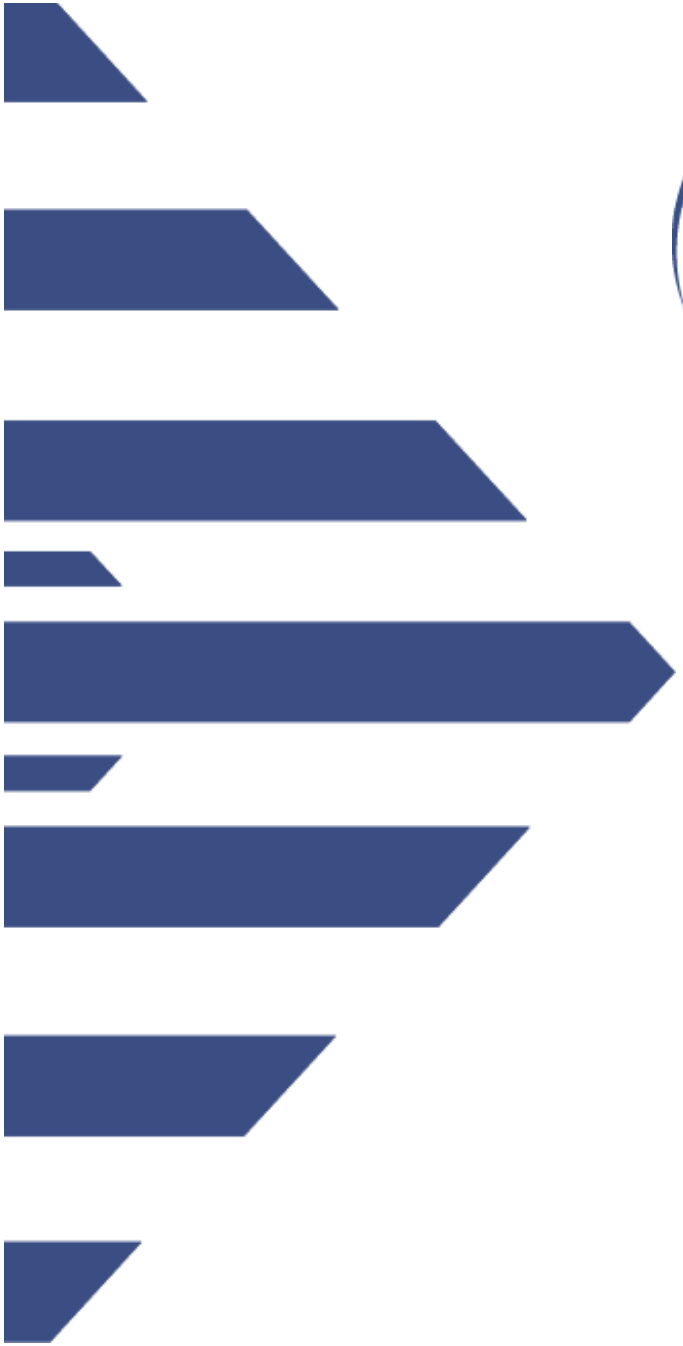


Software Supply Chain Risk Management: From Products to Systems of Systems

Bob Ellison, Chris Alberts, Rita Creel, Audrey Dorofee, and Carol Woody

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
MAY 2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Software Supply Chain Risk Management: From Products to Systems of Systems	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University,Software Engineering Institute,Pittsburgh,PA,15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**29**	

Supply Chains

Supply chain: set of suppliers that contribute to the content of a product or system or that have opportunity to modify its content. (Comprehensive National Cybersecurity Initiative 11)

Hardware product involves multiple deliveries of the same item (built to specification)

Software product is typically a single item redistributed within an organization

Supply-Chain Risk

Hardware supply chains – decades of data collection

- Manufacturing and delivery disruptions
- Manufacturing quality
- Counterfeit hardware estimated at 10%

Software – little data for software supply chains

- Third-party tampering during development or delivery
- Malicious supplier
- Compromised by inadvertent introduction of exploitable design or coding errors

Software Supply Chain Risk Management

Attack Analysis

Factors that lead to successful attacks

Suppliers

Risk-based development

Capability to limit product attributes that enable attacks

Acquirers

Tradeoff decisions between desired use and acceptable business risk

Uncertainty for product/supplier assurance

- limited supply chain visibility and controls
- evolving nature of threats, usage, & product functionality

Continued supply chain risk management during deployment

Attack Example: Stuxnet

Enabled the attacker to modify how the control system managed a physical system. General purpose control systems such as Siemens' execute user supplied software designed for the specific application.

Strategy:

To avoid detection, do not use corporate networks to directly modify the control system software

Use Internet access and defects in Windows or in application software to compromise computing resources belonging to trusted administrators – hundred of thousands of computers were actually compromised. – **Defects are an enabler, and network connectivity is a risk factor.**

Use computing resources such as the USB drives used by system administrators to transfer malware to the control systems **Use of end-user computing resources is a risk factor.**

Use control system extensibility to install control software that would adversely change the behavior of existing control functions. **Product feature is an enabler. No auditing or notification of control code changes are design faults or operational faults.**

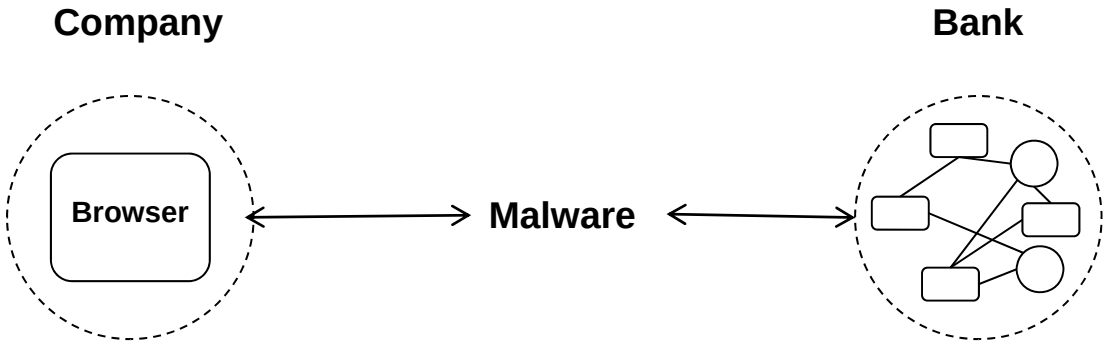
Attack Example: Bank Fraud

Organizations with limited IT support – e.g. school districts

Organization's computer used for bank transaction is compromised

Malware deployed that that receives and can transforms web pages – man-in the middle

When user logs into financial system, a page is returned that informs the user that there will short delay (while malware submits transactions)



Frequent design fault: Financial systems assumed client has not been compromised. Confirmations for fraudulent transactions returned over compromised communications path and blocked by the malware.

Attack Examples

Google: Aurora – access to code base

- Zero-day IE vulnerability
- Social Engineering – targeted employee with access and used chat invitation from “friend” to install malware

RSA: access to sensitive information

- Social engineering
- Flash vulnerability

Epsilon: Access to email addresses

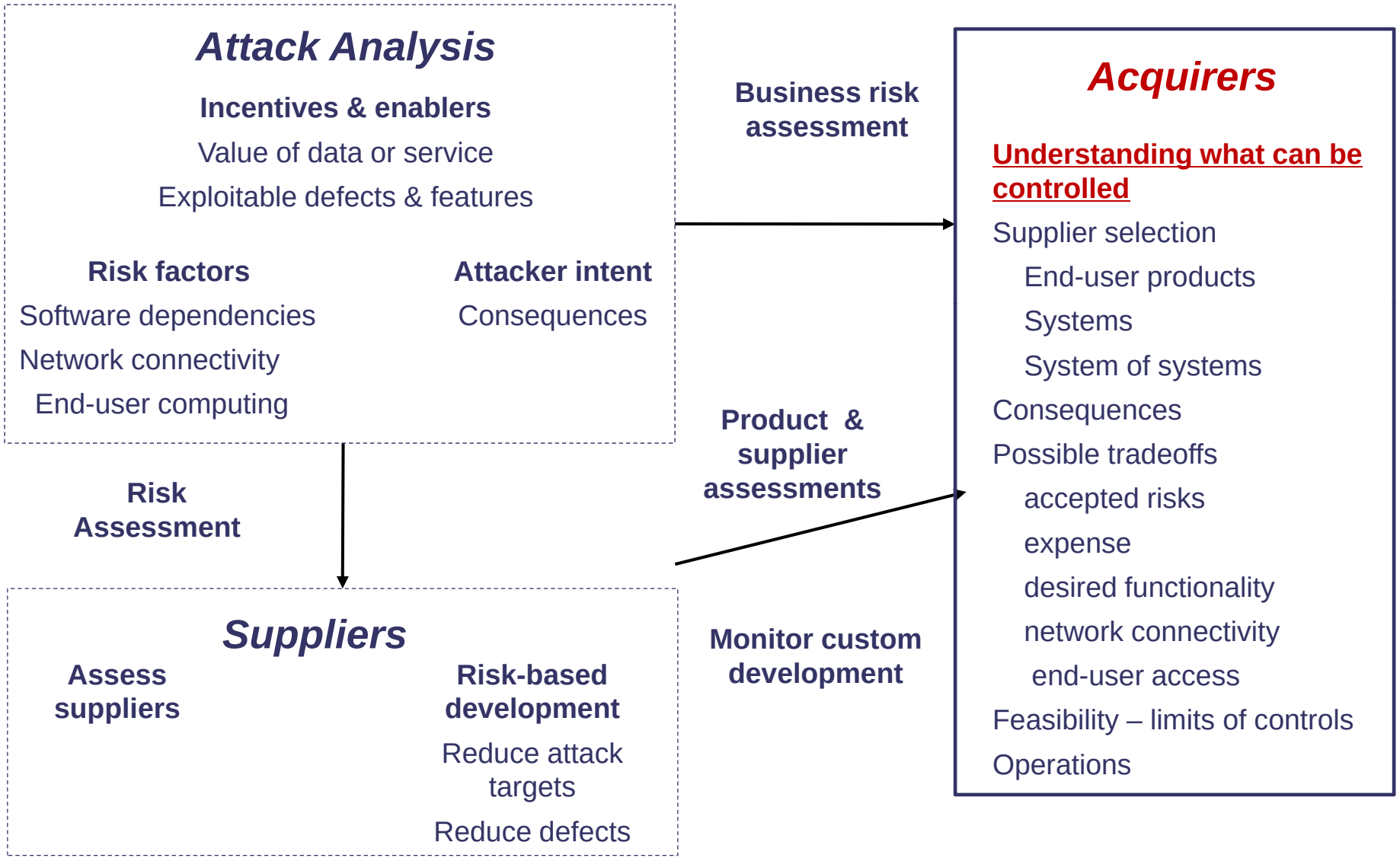
- Social engineering

Changing Nature of Attacks

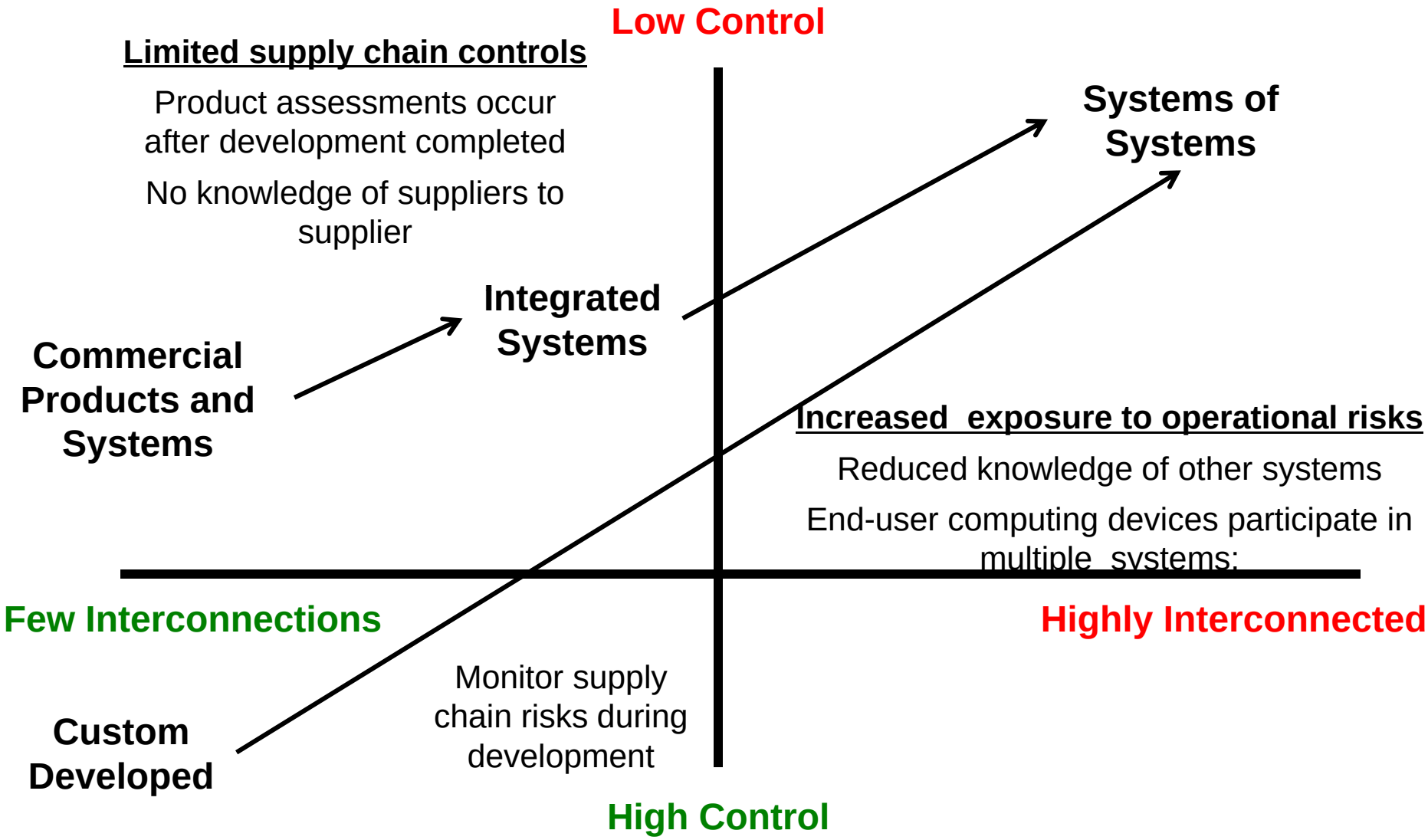
Advanced Persistent Threat (APT)

- Early usage of the term typically focused on the source of the attack such as nation state, organized crime, and terrorist organizations
- After Operation Aurora in 2010 APT became associated with any targeted, sophisticated, or complex attack, regardless of the attacker, motive, origin, or method of operation. [IBM 2010 X-Force Report]

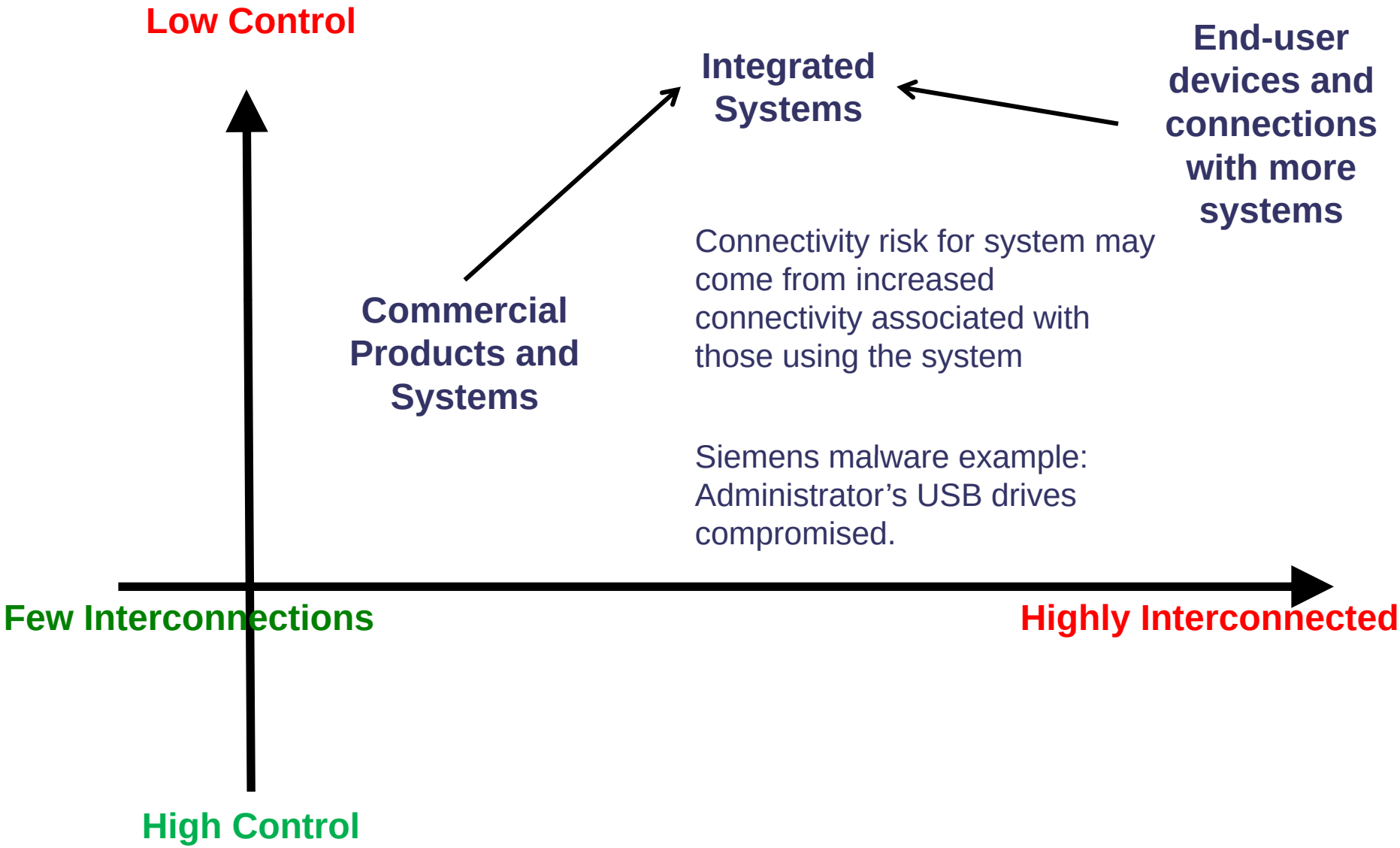
Software Supply Chain Risk Management



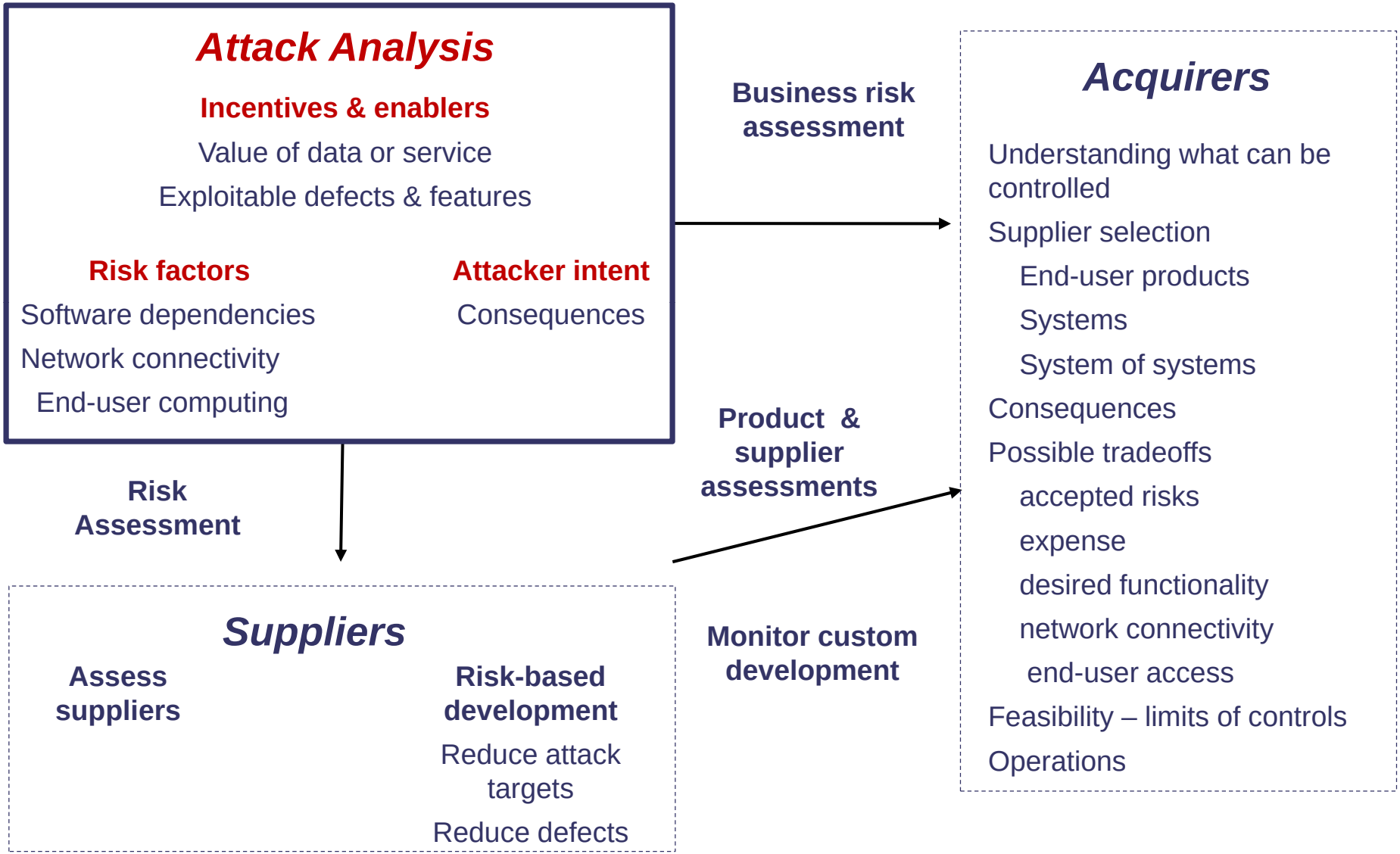
Connectivity and Control[1]



Connectivity and Control[2]



Software Supply Chain Risk Management



Enablers: Software Errors

MITRE has documented software errors that have led to exploitable vulnerabilities: Common Weakness Enumeration (CWE)

CWE/SANS[1] Top 25 Most Dangerous Programming Errors published yearly by MITRE – 3/1/2010

Examples

Improper Input Validation	SQL Injection
Cross-site scripting	Use of Hard-coded Credentials
Download of Code Without Integrity Check	Improper Check for Unusual or Exceptional Conditions
Race Condition	Classic Buffer Overflow

1. http://cwe.mitre.org/top25/
SANS (SysAdmin, Audit, Network, Security) Institute

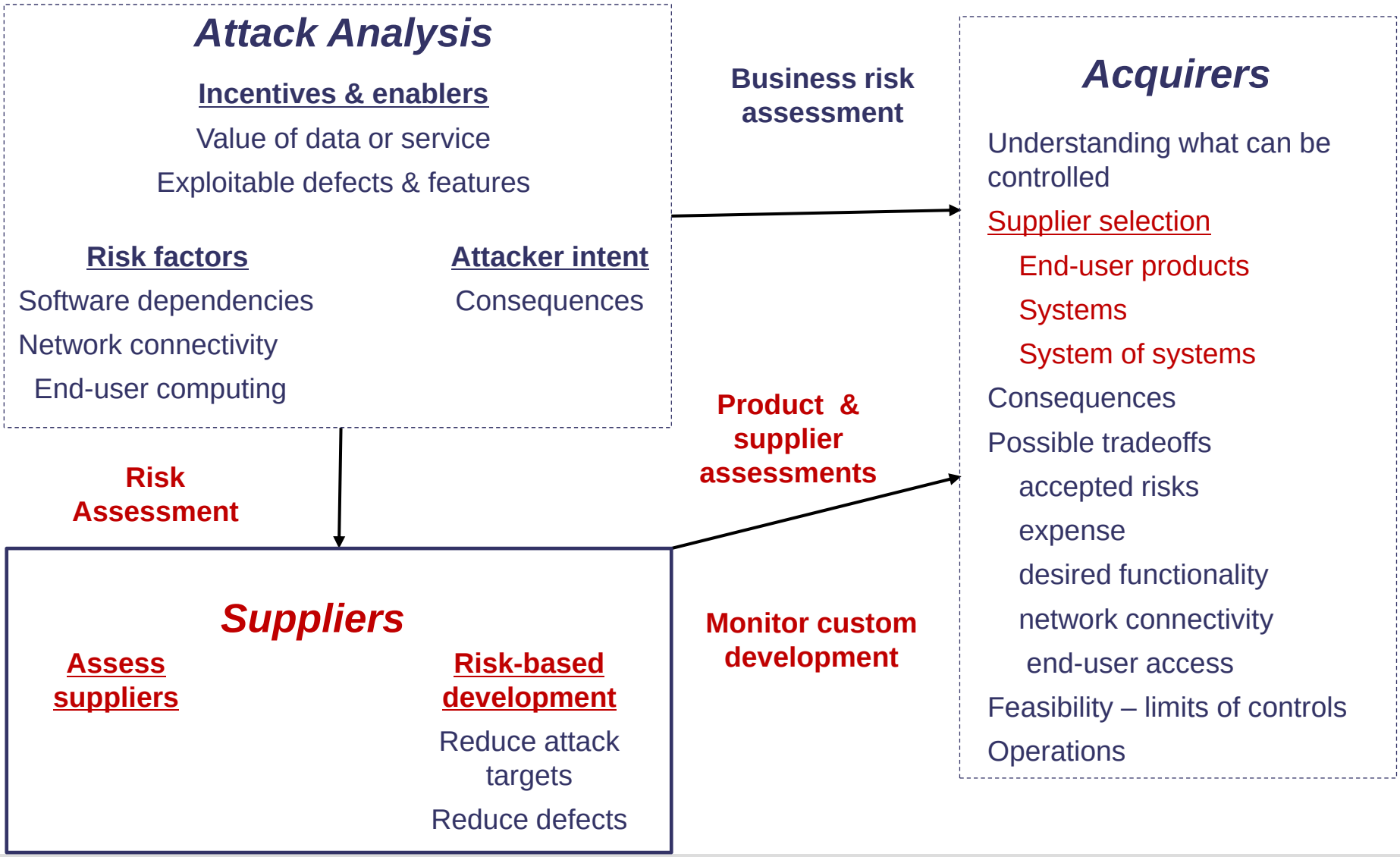
Veracode: State of Software Security

58% of all applications did not achieve an acceptable security score upon first submission Fall 2010

Measured Against CWE/SANS Top-25 Errors

Software Source	Acceptable
Outsourced	6%
Open Source	39%
Internally Developed	30%
Commercial	38%

Software Supply Chain Risk Management



Supplier: Attack Surface Analysis

Reduce Attack Surface

- Remove or change system features or re-architect the implementation to avoid attack enablers or unnecessary channels.
- Revise use of an emerging technology where there is limited knowledge of the potential exploits and mitigations
- Review requirements or implementation if existing mitigations are costly or do not provide the necessary assurance

Supplier: Risk Focused Development

Data flow analysis (threat modeling)

- Consider known weaknesses and attack patterns – e.g. mix of data and commands
- Document security assumptions and trust boundaries
- Consider deployed configuration and expected usage
- Analyze the interfaces to other components (inputs and outputs)
- Consider consequences
- Analyze possible mitigations
- Provide architecture and design guidance
- Guides testing

Secure Software Development

Microsoft: Security Development Lifecycle

Build Security In Maturity Model – http://bsimm.com/

Open Group Trusted Technology Framework for accreditation of technology suppliers – under development with early DoD participation

SafeCode – http://www.safecode.org/

Build-Security-In-Web site – DHS https://buildsecurityin.us-cert.gov/bsi/home.html

General Purpose End-User Software

End-user software has always been a target for attackers

- Floppy disks → Office documents → email → web

Web browser

- Attackers' objective to have user execute their code
  - Extensibility – JavaScript, Ajax, ActiveX
- HTML5 increases browser attack surface

Mobile devices

Software products - systems

Unacceptable risks identified during a product assessment can lead to a rejection – some financial service organization use tests similar to Veracode

Product assessment criteria must reflect the criticality of usage and the level of assurance required.

High	No known failures
Medium	Known vulnerabilities addressed
Low	Failure can be tolerated – low consequence

Open question: Can low assurance components be used in a medium assurance system?

Systems

A systems perspective captures product usage and consequences associated with supply chain risks.

- Changing threat landscape
- Increasing demand for leading-edge software with not well understood risks
- A product's proposed usage and attack opportunities can require mitigations beyond those provided with the product – also applies to legacy systems
- The trust among components implied by the integration
- As we go forward (Cloud Computing) the guidance should be *Don't trust, but verify*[1]

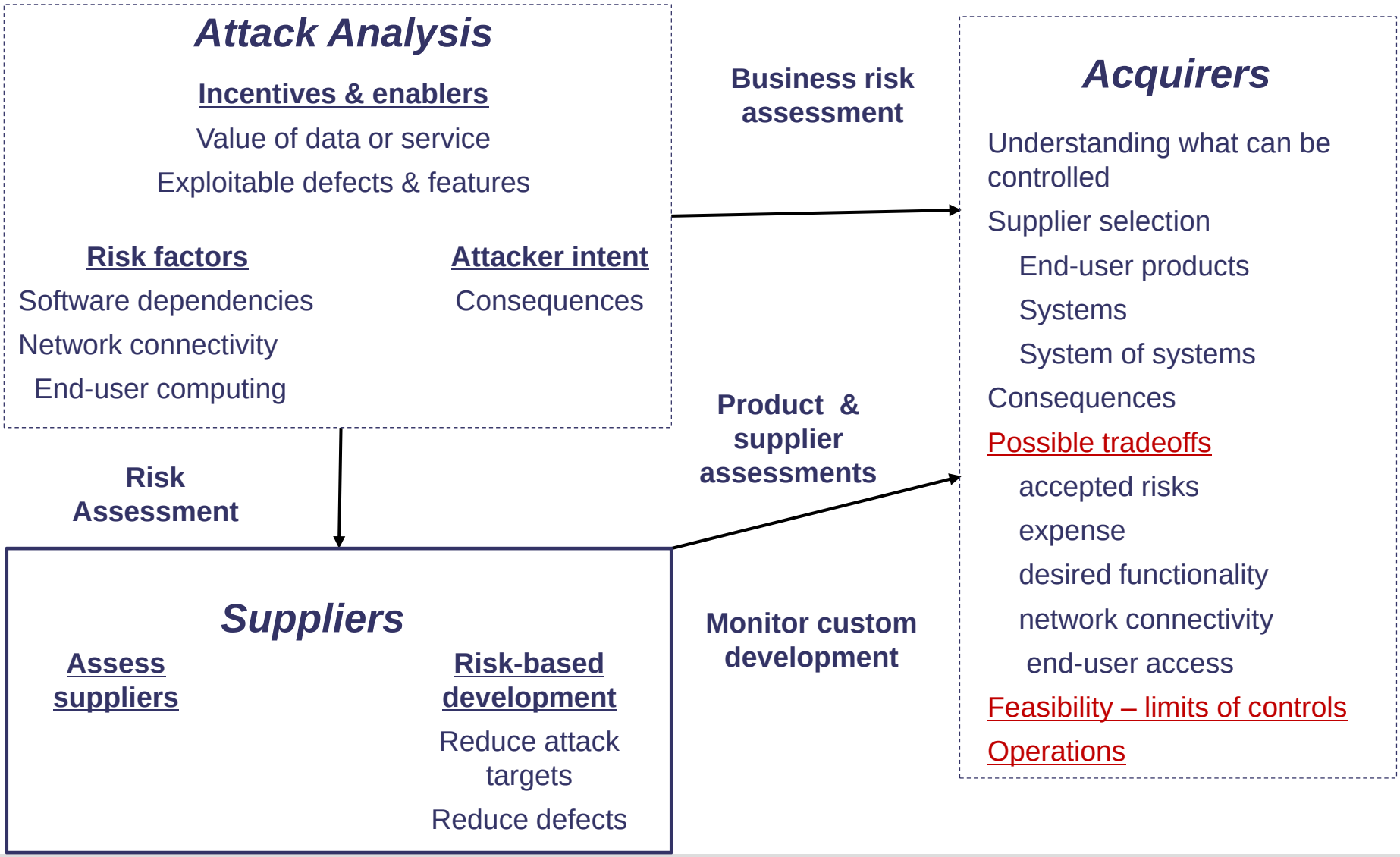
1: Gunnar Peterson, IEEE Security and Privacy, SEPTEMBER /OCTOBER 2010

Stronger Custom Developer Criteria

Applying of practices such as threat modeling at the system level can more demanding than for a product

- Product development
  - Long product life - incremental
  - Concentrate on software weaknesses appropriate to that supplier's domain and products – guided by product history
  - Relatively small and stable set of suppliers
- An integration contractor or custom system developer
  - multiple one-off relatively short-lived efforts
  - multiple functional domains
  - multiple sets of applicable software products, suppliers, and subcontractors

Software Supply Chain Risk Management



Trade-Offs

A simplified design to reduce cost or speed delivery may not provide adequate mitigations for known operational risks.

Products that support end-user runtime customization can provide that same capability to an attacker.

The use of emerging technologies with exploits that are not well understood increases risk.

System functionality may have to be changed or a higher risk accepted if mitigation costs for a desired feature are too high or if residual risks for known mitigations are higher than anticipated.

Supply Chain Control Limitations

Total prevention is not feasible because of the sheer number of risks; limited supply chain visibility; uncertainty of product assurance; and evolving nature of threats, usage, and product functionality

Defense-in-depth does not necessarily reduce risks – we often do not understand interactions among multiple mitigations.

Operations Over Time

Supply chain risk mitigation is not a one-time event

- New attack techniques and software weaknesses may be discovered.
- Product upgrades that add features or change design can invalidate the results of prior risk assessments and may introduce vulnerabilities.
- Corporate mergers, new subcontractors, or changes in corporate policies, staff training, or software development processes may eliminate expected SCRM practices.
- Product criticality may increase with new or expanded usage.

Summary

Increased connectivity and interoperability raise the value of considering supply chain risks for *secondary* applications.

Techniques exist to reduce occurrence of software vulnerabilities but are not yet widely applied.

A systems perspective, particularly in deployment, captures product usage and consequences associated with supply chain risks.

- Component update or replacement
- Change in usage
- Evolving threats

Sources

Software Supply Chain Risk Management: From Products to Systems of Systems

- http://www.sei.cmu.edu/library/abstracts/reports/10tn026.cfm

Evaluating and Mitigating Software Supply Chain Security Risks

- **http://www.sei.cmu.edu/library/abstracts/reports/10tn016.cfm**

Attack Surface

- Michael Howard, 2003, http://msdn.microsoft.com/en-us/library/ms972812.aspx

Threat Modeling

- Frank Swiderski, Window Snyder, *Threat Modeling*, 2004
- Michael Howard and Steve Lipner. *The Security Development Lifecycle,* 2006
- James McGovern, & Gunnar Peterson. “10 Quick, Dirty, and Cheap Things to Improve Enterprise Security.” Security & Privacy, IEEE, March-April 2010
- Building Security In Maturity Model (BSIMM) http://bsimm2.com/index.php
- John Stevens, “Threat Modeling— Perhaps It’s Time”, Security & Privacy, IEEE, May-June 2010